An Execution Analyzer for the Sigma 5 Computer

C. C. Klimasauskas
Communications Systems Research Section

Since many different computers on the market today claim to perform the same class of tasks, and no uniform criterion has been established to aid the decision-making process, the problem of how to select the "best" computer for a particular job mix becomes almost a matter of personal preference and chance. One technique for evaluating the performance of a central processing unit is based on the frequency of usage of the various machine instructions. This technique is particularly applicable to machines used in a dedicated process-control activity such as those of the DSN Tracking Stations. This article describes a program which has been written for the Sigma 5 computer to gather data on the dynamic usage of computer instructions in various tasks.

I. Introduction

The technique for evaluating the performance of a central processing unit based on the frequency of usage of the various machine instructions is particularly applicable to machines used in a dedicated process-control activity such as those of the DSN Tracking Stations. This article describes a program written for the Sigma 5 computer to gather data on the dynamic usage of computer instructions in various tasks. Results of a study using this program are described in a companion article by Layland and Klimasauskas[1].

This data gathering program is named MRφφ7, and its overall operation is described in Section II. Subsequent sections include instructions for MRφφ7 operation, and for attaching it to various types of user programs; a description of the format in which the data gathered are written into a disk file; and a description of error conditions which may be encountered, and their associated messages. Section V describes in some detail the set of programs which have been used to transform the data gathered by MRφφ7 into a visually meaningful format, and presents an example of that output format.

II. The Interpreter

MRφφ7 is an interpretive executor which dynamically tabulates instruction usage. The entire problem-solving repertoire of Sigma 5 instructions is allowed.[2] Results from program execution are stored in a disk file for further analysis. Instructions which communicate with the monitor (CALs) are limited to background mode only. Monitor CALs dealing with interrupts, program control transfers, and traps are excluded.

MRφφ7 may be tailored to operate in either a Batch Processing Monitor (BPM) or a Batch Timesharing Monitor (BTM) environment through an assembly control parameter. This parameter (P:USER), when set equal to :BPM, produces code to interpret BPM monitor CALs.

[1]Layland, J. W., and Klimasauskas, C. C., *A Myopic View of Computer-Based System Design* (to be published).

[2]These do not include control, and input/output instructions.

When set equal to :BTM, it produces code to interpret BTM monitor CALs.

Since this program was designed as a tool to study a wide range of problems, certain features (described in Section VI) were not considered of sufficient importance to warrant implementation. If these or other features (such as foreground CALs) must be used, their code may be bypassed through the use of a HOLD-GO pair (see Section III).

To facilitate analysis of data from MRφφ7, a series of programs have been written which transform the data into informative formats. These are described in Section V.

III. Operating Instructions

A. FORTRAN Programs

MRφφ7 is easily attached to any extended FORTRAN IV program. To activate it, simply use this statement:

```
CALL MRφφ7
  :
STOP
```

If it is desired to deactivate MRφφ7 with intentions of restarting it, use the following sequence:

```
CALL MRφφ7
     *    }  Program executed under control of
     *    }  MRφφ7
     *
CALL HOLD
     .    }  Program executed without control of
     :    }  MRφφ7
CALL GO
     *    }  Program executed under control of
     *    }  MRφφ7
     *
STOP
```

MRφφ7 also provides the option of writing its current data tables to M:EO, clearing them, and relinquishing control of the program. This effectively provides the user the capability of selectively looking at the instruction repertoire of several very different portions of the program. A word of caution: MRφφ7 *must* be in control of the program when it exits to the monitor. If MRφφ7 is not in control when the program is to be terminated, the data file will be lost. To close and save the output file, use "CALL WDATCLSE." This routine will exit to the monitor on closing the file. Note that all user output files which are open will be lost. A sample sequence is:

```
CALL MRφφ7
     *    }  Program executed under MRφφ7
     *    }  control
     *
CALL MRφφ7
     .    }  Data tables written to
     .    }  M:EO ("DEXEC"), cleared, and
     .    }  control relinquished
     .    }
     .    }  Program executed without control
     .    }  of MRφφ7
CALL MRφφ7
     *    }  Program executed under MRφφ7
     *    }  control
     *
CALL MRφφ7
     .    }  Data tables written to
     .    }  M:EO ("DEXEC"), cleared, and
     .    }  control relinquished
CALL MRφφ7
STOP      }  End sequence giving MRφφ7
          }  control
```

It is allowable to use any number of HOLD-GO pairs within a MRφφ7 pair.

B. Assembly Programs

MRφφ7 is attached to assembly programs using the assembly language equivalent of the FORTRAN calling sequences above, namely:

```
  CALL MRφφ7
is replaced by
    REF MRφφ7
    LI,14  0
    BAL,15 MRφφ7

  CALL GO
is replaced by
    REF GO
    LI,14  0
    BAL,15   GO

  CALL HOLD
is replaced by
    REF HOLD
    LI,14  0
    BAL,15 HOLD

  STOP
is replaced by
    M:EXIT
```

IV. Data Record Format

MRϕϕ7 writes the data tables to a disk file ("DEXEC") through the M:EO DCB. The DCB does not need to be assigned through an ASSIGN card, nor should it be. For each MRϕϕ7 pair and for the MRϕϕ7-STOP pair, a data record is written. If data already exist in file "DEXEC," they will be lost. The format of each record is as shown in Fig. 1.

V. Support Programs

To facilitate the analysis of data from MRϕϕ7, a number of support programs have been written to transform the data to usable and meaningful terms. These programs are briefly described below.

A. PUNCH

PUNCH is a FORTRAN program which takes the file DEXEC produced by MRϕϕ7 and generates records in the standard binary format acceptable to FORTRAN IV. It uses an assembly language subroutine RDATA to read the data blocks produced by MRϕϕ7. Figure 2 describes the actions of PUNCH.

After each logical record is written, a !EOD is written. After the last logical record, two !EODs are written (an EOF).

B. Subroutine GETREC

GETREC is a FORTRAN subroutine which reads records produced by PUNCH into a data array. It tests sense switch 1 to determine the source device. The calling sequence is

```
CALL GETREC  (IDAT(1), IDATL,999S)
```

when

IDAT(1) is the first element of the storage area. IDATL is the number of words to be read into the data storage area (currently 545). 999S is the address to which the routine will return when an end of all data (a double !EOD on the card reader, or an EOF on the DISK) is encountered.

If sense switch 1 is reset,

Data will be read from the Card Reader through F:8.

If sense switch 1 is set,

Data will be read from the DISK (DPEXEC) through F:7.

Figure 3 is a diagramatic overview of GETREC.

C. ACCTNG

ACCTNG is a FORTRAN program which produces a graphical summary of the execution data. The summary[3] consists of the following sections:

(I) Complete execution summary—*All* data are displayed.

(1) Instruction usage summary (Fig. 4)
The bar graph for the execution of an instruction shown in Fig. 4 is relative to the most used instruction. The bar graphs for indirect addressing (*), indexing (X), and indexed indirect addressing (*X) are relative to 100%. Unused instructions and addressing modes are not shown.

(2) Register usage summary
REGISTER USAGE refers to use of a register as an operand. INDEX REGISTER USAGE refers to use of a register for indexing. Bar graphs in each section are relative to the most used register in that section, as may be seen in Fig. 5.

(3) Execution classes
This is a one-page summary (see Fig. 6) of instruction usage. It is self-explanatory. Instructions are listed by their standard mnemonics.

(II) Condensed summary—Selected data are displayed

(1) Instruction usage summary
The condensed instruction usage summary is almost identical to the instruction usage summary. However, only instructions used more than 0.9% are listed, and the bar graphs for indirect, indexed, and indexed indirect addressing are relative to the individual instruction usage (see Fig. 7).

(2) Instruction overview
The instruction overview is a bar graph on a logarithmic scale of instruction usage. Instruc-

[3]The example used in this section is the execution of an optimized assembly language, FFT, written by Dr. Howard Rumsey. The FFT was executed twice in succession on a 256-point complex-element array.

tion mnemonics are on the bottom of the graph, and the corresponding usage is above. If an instruction was used at all, though not enough to appear on the graph, it has a "#" on the 1% grid. The total number of instructions is displayed on the 100% grid (see Fig. 8).

An instruction overview may be very revealing. Figure 8 shows an overview for an optimized assembly language FFT. Figure 9 shows an overview for an optimized FORTRAN IV FFT. Both were used to do a double transform on a 256-point complex-data array, and both were written by a highly skilled programmer. The FORTRAN program required almost twice as many instructions to do the transform as the assembly program. However, the absolute usage of floating point instructions is about the same (9–15% for assembly, 2 X 5–7% for FORTRAN).

ACCTNG will suppress all but the instruction overview if sense switch 2 is set. ACCTNG gets data through the subroutine GETREC (described in Section VB). Additionally, ACCTNG requires an index file to give it the op codes and mnemonics of the instructions, and control information for grouping the instructions together into logical blocks. The format of records in this file is as follows:

(1) Op code, 4-letter mnemonic (blanks are significant)

(2) Op code | sub code, 4-letter mnemonic

(3) O, blanks

In this format,

Item (1) is the standard op code–mnemonic pair for an instruction. For example, the STORE REGISTER instruction, op code X'35', would be represented as "35,STW."

Item (2) is for instructions which are actually a subclass of another instruction. In this case, the two-digit location of the data for the subcode is concatenated onto the two-digit op code. For example, the unconditional branch instruction (BCR,O) has data describing it stored in op code 01, and is represented as "6801,B."

Item (3) establishes the end of a logical instruction group.

A diagrammatic summary of ACCTNG is shown in Fig. 10.

VI. Error Messages and Other Miscellany

The following is a brief description of error messages and a possible explanation as to why they may occur, followed by a discussion on some conditions which have not been implemented and may cause program failure.

MRφφ7 intercepts illegal and unimplemented instructions and gives a warning accordingly. MRφφ7 also aborts on CALs that are the object of an EXU instruction, and flags EXU instructions which are nested. These events are accompanied by an error dump of the format shown below.

```
   ABORT AND WARNING FORMAT:
             HHHHHHHH
Rφ,RBLKφ
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
Rφ,RBLK1
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             RRRRRRRR  RRRRRRRR  RRRRRRRR  RRRRRRRR
             LLLLLLLL  IIIIIIII  EEEEEEEE  CCφφφφφφ
```

when

RBLKϕ	— Register block zero → the program register block.
RBLK1	— Register block one → MRϕϕ7 register block (or pseudo-register block).
RRRRRRRR	— Hexadecimal contents of the register.
LLLLLLLL	— Location of offending instruction.
IIIIIII	— Offending instruction.
EEEEEEEE	— SR3 from the last CAL 1 which aborted or errored.
CCϕϕϕϕϕϕ	— BYTE ϕ contains the current condition code and floating control.

ERROR HEADERS ("HHHHHHHH" from above)

"ABRT CAL"	— A CAL without an ERR/ABN exit has returned through the ERR/ABN exit with an abort level error code. — Abort.
"BAD CAL"	— An illegal CAL or a CAL with an illegal FPT code has been encountered. — Abort.
"PRIV INS"	— An attempt has been made to execute a privileged instruction. — Abort.
"DATA INS"	— An illegal/unimplemented instruction has been encountered. This includes CAL2 and CAL4 in BTM and CAL2, CAL3, CAL4 in BPM. — Abort.
"EXU CAL"	— An attempt to execute a CAL has been made. — Abort.
"EXU EXU"	— More than one level of EXU instruction. — Warning.

Miscellaneous Information

FPTs which are longer than 40 words will be truncated to 40 words. If there are any FPTs in the user program longer than 40 words, they should be reduced to 40 words or less.

SNAP CALs inserted by the monitor have their FPTs in 01 protection type. Since MRϕϕ7 currently modifies the FPT, the monitor will abort on a memory protection violation when the SNAP is encountered. Therefore, !SNAP cards cannot be used in a region of the program to be executed by MRϕϕ7. (M:SNAP FPTs are in 00 protection-type memory, and may be used.)

Since MRϕϕ7 uses the M:EO DCB for saving data, this DCB must not be used by the program. Further, to insure that the file (DEXEC) written through the M:EO DCB is not destroyed, the user must

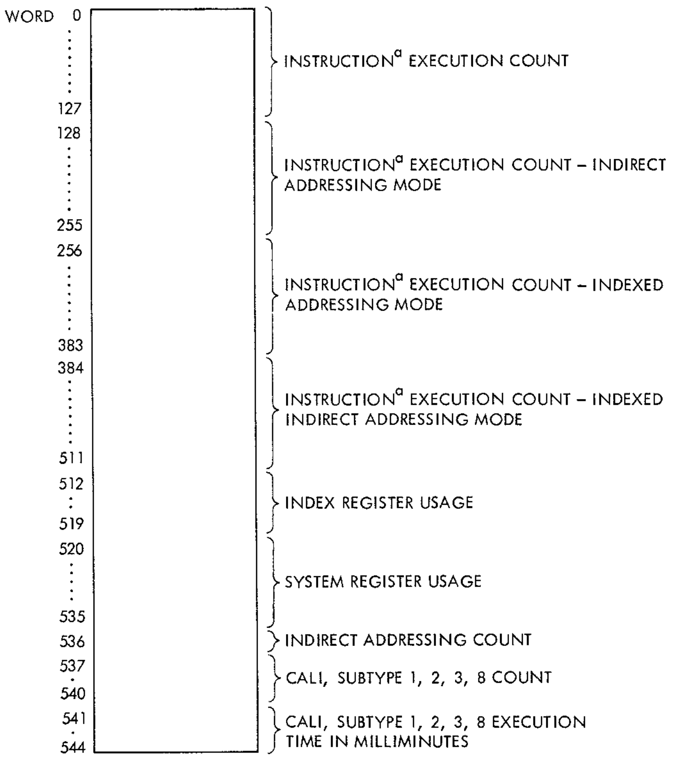
(1) exit to the monitor (through an M:EXIT, M:XXX, or M:ERR) while under the control of MRϕϕ7

or

(2) exit to the monitor via

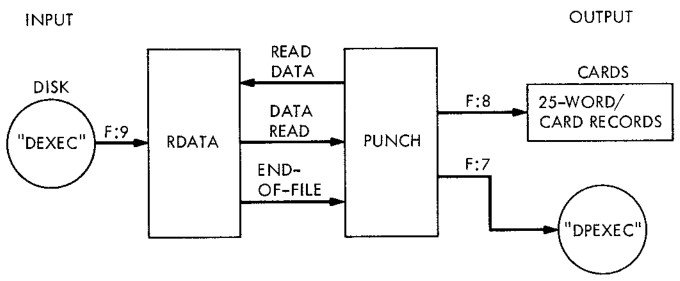
```
    CALL  WDATCLSE
or
    REF   WDATCLSE
    B     WDATCLSE
```

This closes the M:EO DCB with SAVE. All unclosed user output files are lost.


WORD 0
127
128
255
256
383
384
511
512
519
520
535
536
537
540
541
544
INSTRUCTION[a] EXECUTION COUNT
INSTRUCTION[a] EXECUTION COUNT - INDIRECT ADDRESSING MODE
INSTRUCTION[a] EXECUTION COUNT - INDEXED ADDRESSING MODE
INSTRUCTION[a] EXECUTION COUNT - INDEXED INDIRECT ADDRESSING MODE
INDEX REGISTER USAGE
SYSTEM REGISTER USAGE
INDIRECT ADDRESSING COUNT
CALI, SUBTYPE 1, 2, 3, 8 COUNT
CALI, SUBTYPE 1, 2, 3, 8 EXECUTION TIME IN MILLIMINUTES

[a]AN ADDITIONAL INSTRUCTION SUBCLASS (BCR, O) HAS BEEN ASSIGNED OP CODE X'01'.

Fig. 1. Data record format for DEXEC



F:7, F:8, F:9 MUST ALL BE ASSIGNED AT LOAD AND EXECUTION TIME.

Fig. 2. Overview of PUNCH

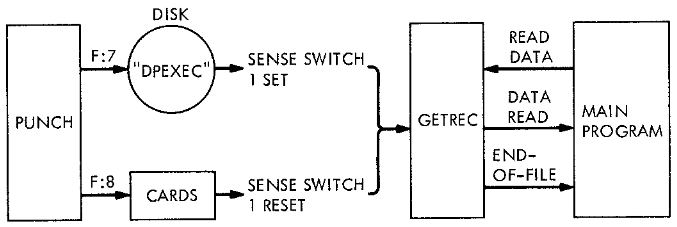


Fig. 3. Operational diagram of GETREC

```
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<                                                                                                                >
<                                  ********** CØMPLETE SUMMARY ********** RECØRD # 1                            >
<                                                                                                                >
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>


 INSTRUCTIØN USAGE SUMMARY -- GRAPHICAL
 ØP - MNEUMØNIC - REL USAGE - ABS USAGE
 INDIRECT ADDRESSING! REL USAGE - ABS USAGE
 INDEXING! REL USAGE - ABS USAGF
 INDIECT AND INDEXING! REL USAGF - ABS USAGE


 22•LI        1.690%,    1075!*******

 72•LB         .015%,       9!
        *    88.89%,       8!**************************************************************************
        X    11.11%,       1!********

 32•LW        6.825%,    4343!********************************
        *    24.09%,     1046!********************
        X    24.27%,     1054!********************
        *X   23.88%,     1037!********************

 12•LD       13.677%,    8703!****************************************************************
        *    47.06%,     4096!**************************************
        X    52.92%,     4606!********************************************
        *X   47.06%,     4096!**************************************

 3B•LAW        .202%,     128!

 75•STB        .004%,       2!
        X   100.00%,       2!***********************************************************************************

 35•STW       4.325%,    2752!********************
        *    37.21%,     1024!******************************
        X    37.57%,     1034!*******************************
        *X   37.21%,     1024!******************************

 15•STD       6.690%,    4257!***********************-*******
        *    96.22%,     4096!*******************************************************************************
        X    99.23%,     4224!**********************************************************************************
        *X   96.22%,     4096!*******************************************************************************
##############################

 20•AI        2.287%,    1455!**********

 30•AW        2.025%,    1288!*********

 10•AD        3.244%,    2064!***************

 38•SW         .002%,       1!

 37•MW         .005%,       3!

 36•DW         .030%,      19!
##############################

 73•MTB        .406%,     258!*
##############################

 21•CI         .804%,     511!***

 31•CW        6.689%,    4256!*******************************
        X      .19%,        8!

 19•CLM        .203%,     129!
##############################
```

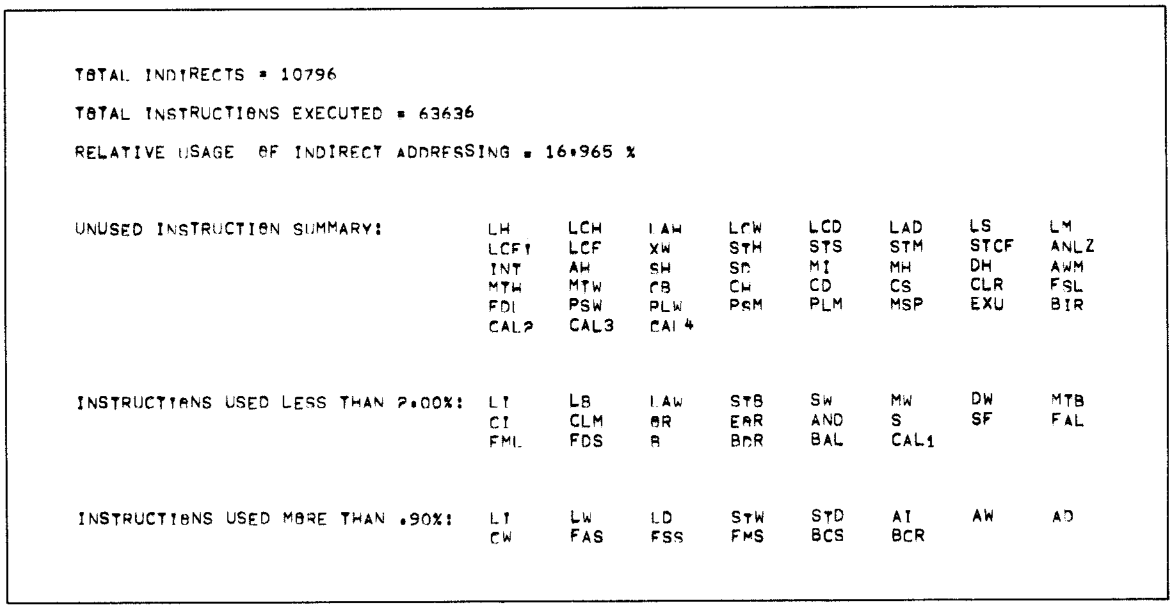
Fig. 4. Complete summary format from ACCTNG

```
 49*BR         *004%,       2!

 48*EBR        *610%,     388!**

 4B*AND        *007%,       4!

##############################

 25*S          *827%,     526!***

 24*SF         *406%,     258!*

##############################

 3D*FAS     10*556%,     6717!****************************************************

 1D*FAL        *398%,     253!*

 3C*FSS      9*952%,     6333!*************************************************

 3F*FMS     17*489%,    11129!**************************************************************************************
          *     4*58%,    510!***

 1F*FML        *398%,     253!*

 3E*FDS        *004%,       2!

##############################

 69*BCS      5*854%,     3725!***************************

 68*BCR      3*536%,     2250!*****************
          *      *71%,     16!
          X   23*29%,     524!*******************
          *X     *09%,      2!

 01  B         *846%,     538!***
          *     2*97%,     16!**
          X   97*21%,     523!********************************************************************************
          *X     *37%,      2!

 64*BDR        *002%,       1!

 6A*BAL        *848%,     539!***

##############################

 04*CAL1       *005%,       3!

##############################

 CAL1,1!    *008MIN,      2'S ***************************************************************************************

 CAL1,2!    *000MIN,      0'S

 CAL1,3!    *000MIN,      0'S

 CAL1,8!    *000MIN,      0'S
```

Fig. 4. (contd)

```
REGISTER USAGE:

0 =   .000%,        0:
                     :

1 =  1.877%,     1077:*****
                     :*****

2 =  8.987%,     5158:***************************
                     :***************************

3 =  1.795%,     1030:*****
                     :*****

4 =  7.999%,     4591:************************
                     :************************

5 =   .004%,        2:
                     :

6 =   .915%,      525:**
                     :**

7 =  1.784%,     1024:*****
                     :*****

8 = 25.990%,    14918:******************************************************************************
                     :******************************************************************************

9 =  9.556%,     5485:*****************************
                     :*****************************

A = 17.844%,    10242:******************************************************
                     :******************************************************

B =  7.152%,     4105:**********************
                     :**********************
C = 10.704%,     6144:********************************
                     :********************************

D =  3.581%,     2055:***********
                     :***********

E =   .901%,      517:**
                     :**

F =   .919%,      527:**
                     :**

 INDEX RFGISTFR USAGF:

1 =  4.646%,      532:**********
                     :**********

2 = 35.816%,     4102:******************************************************************************
                     :******************************************************************************

3 = 35.851%,     4106:******************************************************************************
                     :******************************************************************************

4 =  1.197%,      137:**
                     :**

5 =   .018%,        2:
                     :

6 =  4.593%,      526:**********
                     :**********

7 = 17.882%,     2048:***************************************
                     :***************************************
```

Fig. 5. Register usage summary format from ACCTNG



```
TØTAL INDIRECTS = 10796

TØTAL INSTRUCTIØNS EXECUTED = 63636

RELATIVE USAGE  ØF INDIRECT ADDRESSING = 16.965 %


UNUSED INSTRUCTIØN SUMMARY:          LH      LCH     LAW     LCW     LCD     LAD     LS      LM
                                     LCFI    LCF     XW      STH     STS     STM     STCF    ANLZ
                                     INT     AH      SH      SD      MI      MH      DH      AWM
                                     MTH     MTW     CB      CH      CD      CS      CLR     FSL
                                     FDL     PSW     PLW     PSM     PLM     MSP     EXU     BIR
                                     CAL2    CAL3    CAL4


INSTRUCTIØNS USED LESS THAN 2.00%:   LI      LB      LAW     STB     SW      MW      DW      MTB
                                     CI      CLM     ØR      EØR     AND     S       SF      FAL
                                     FML     FDS     B       BDR     BAL     CAL1


INSTRUCTIØNS USED MØRE THAN .90%:    LI      LW      LD      STW     STD     AI      AW      AD
                                     CW      FAS     FSS     FMS     BCS     BCR
```

Fig. 6. Instruction class summary from ACCTNG

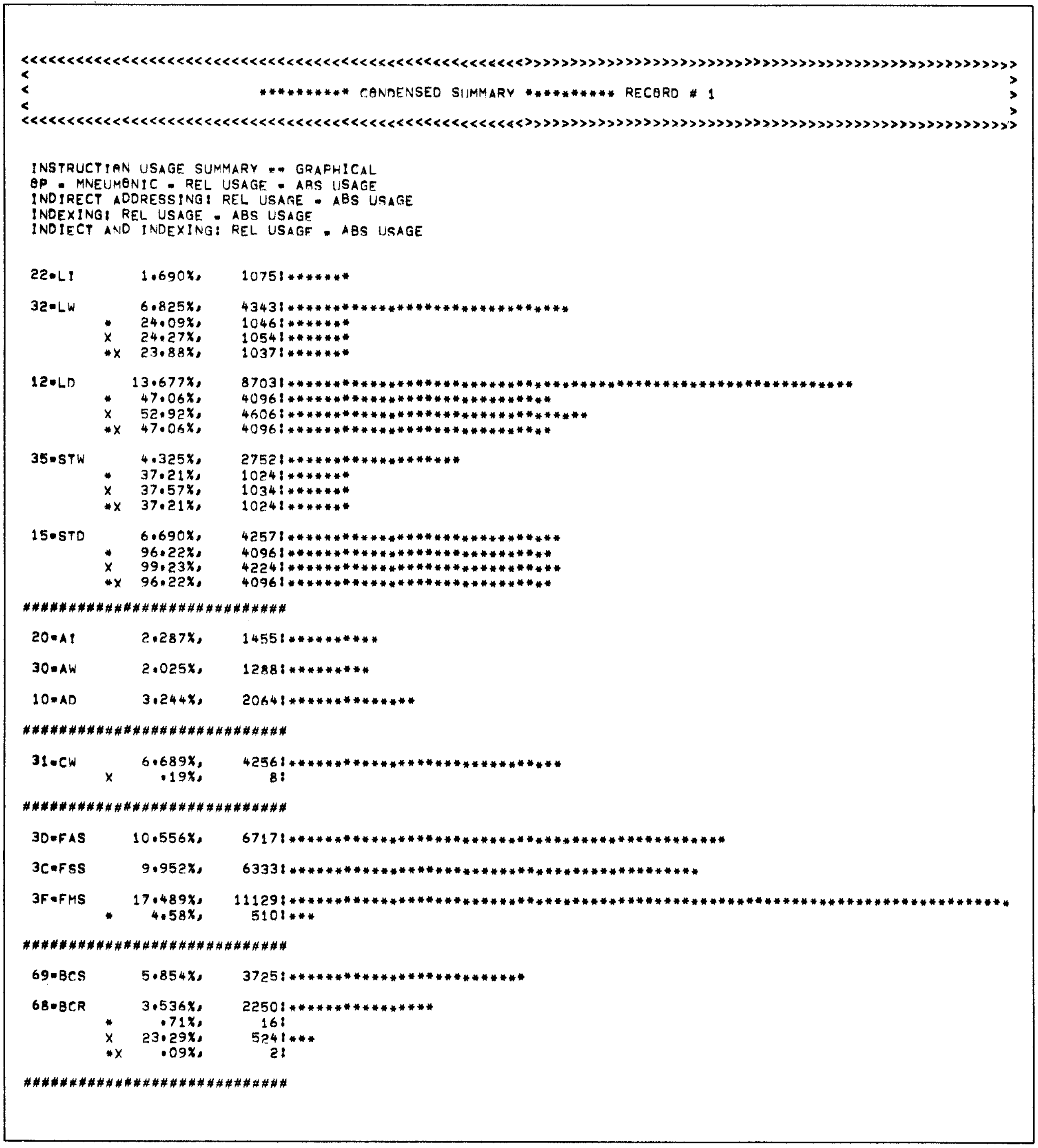

```
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<                                                                                                             >
<                         ********** CONDENSED SUMMARY ********** RECORD # 1                                  >
<                                                                                                             >
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>


 INSTRUCTION USAGE SUMMARY -- GRAPHICAL
 OP - MNEUMONIC - REL USAGE - ABS USAGE
 INDIRECT ADDRESSING! REL USAGE - ABS USAGE
 INDEXING! REL USAGE - ABS USAGE
 INDIECT AND INDEXING: REL USAGF - ABS USAGE


 22•LI        1•690%,    1075!*******

 32•LW        6•825%,    4343!*****************************
          *  24•09%,     1046!*******
          X  24•27%,     1054!*******
          *X 23•88%,     1037!*******

 12•LD       13•677%,    8703!**********************************************************
          *  47•06%,     4096!***************************
          X  52•92%,     4606!*******************************
          *X 47•06%,     4096!***************************

 35•STW       4•325%,    2752!******************
          *  37•21%,     1024!*******
          X  37•57%,     1034!*******
          *X 37•21%,     1024!*******

 15•STD       6•690%,    4257!****************************
          *  96•22%,     4096!***************************
          X  99•23%,     4224!****************************
          *X 96•22%,     4096!***************************
#############################

 20•AI        2•287%,    1455!**********

 30•AW        2•025%,    1288!*********

 10•AD        3•244%,    2064!**************
#############################

 31•CW        6•689%,    4256!****************************
          X     •19%,       8!
#############################

 3D•FAS      10•556%,    6717!*********************************************

 3C•FSS       9•952%,    6333!******************************************

 3F•FMS      17•489%,   11129!**************************************************************************
          *   4•58%,      510!***
#############################

 69•BCS       5•854%,    3725!*************************

 68•BCR       3•536%,    2250!***************
          *     •71%,      16!
          X  23•29%,      524!***
          *X    •09%,       2!

#############################
```

Fig. 7. Condensed summary format from ACCTNG

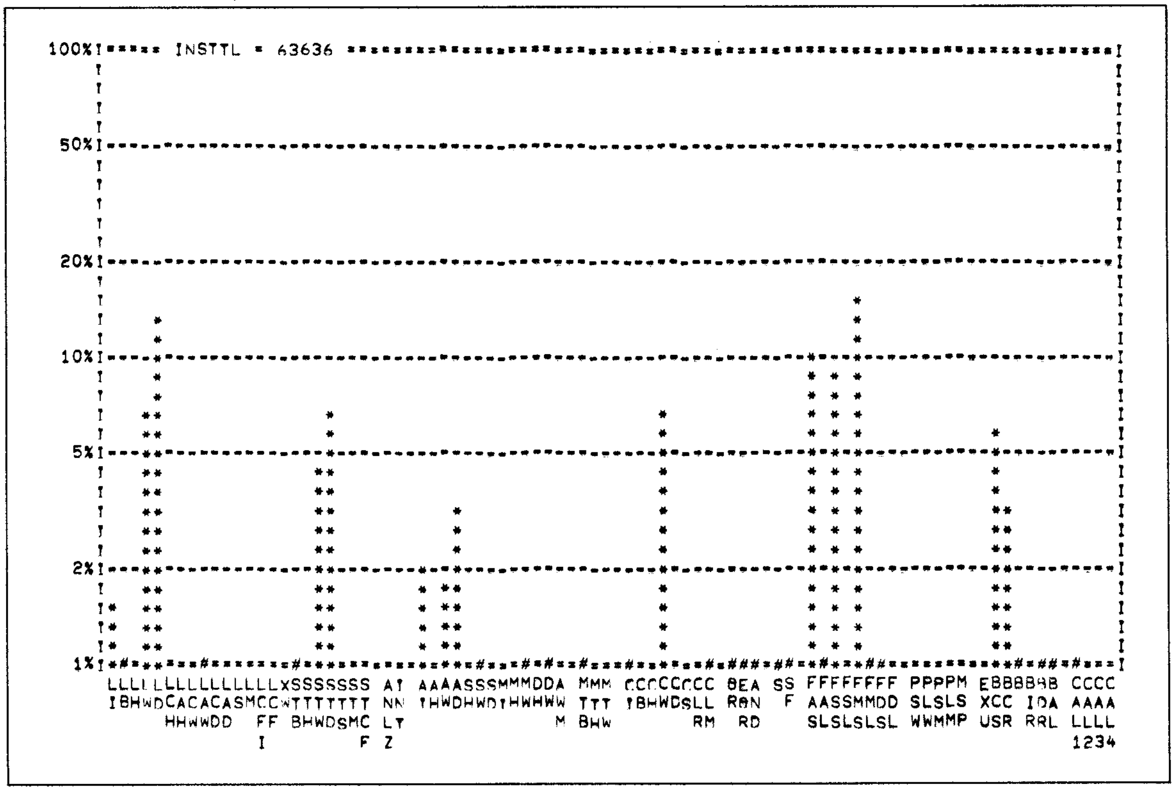


Fig. 8. Instruction overview of assembly-language FFT

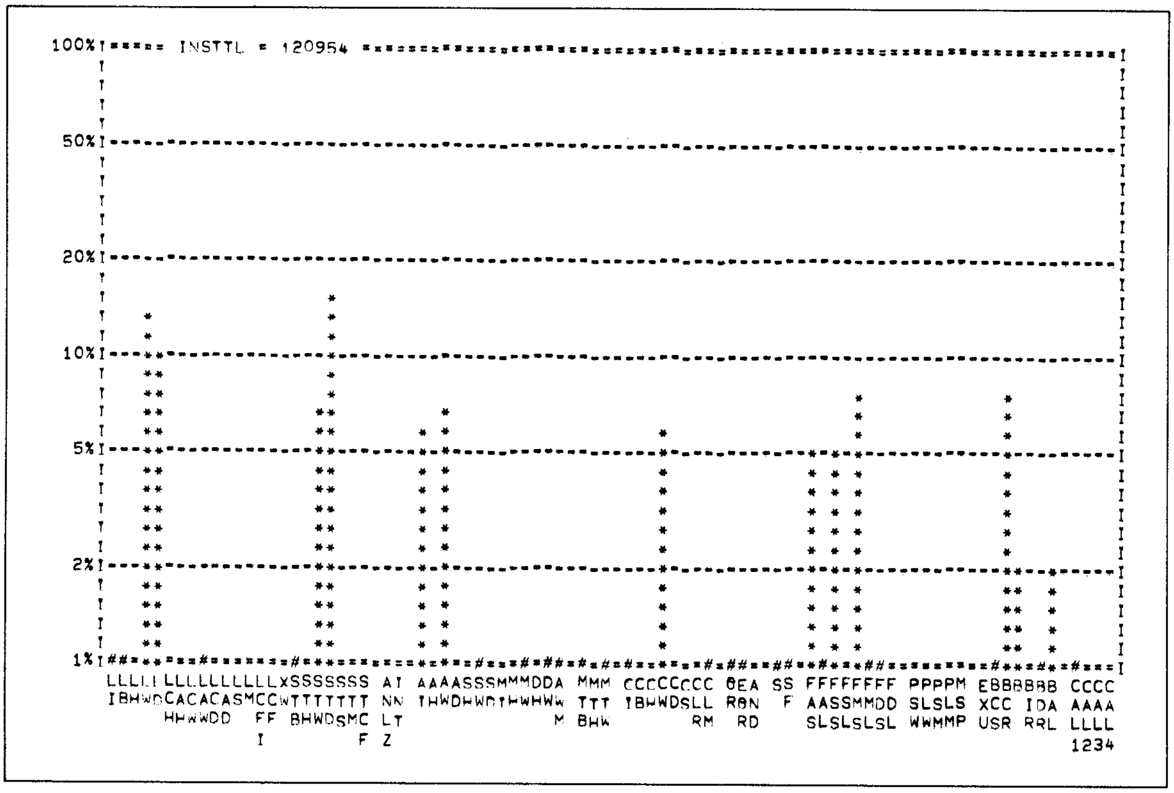


Fig. 9. Instruction overview of FORTRAN FFT

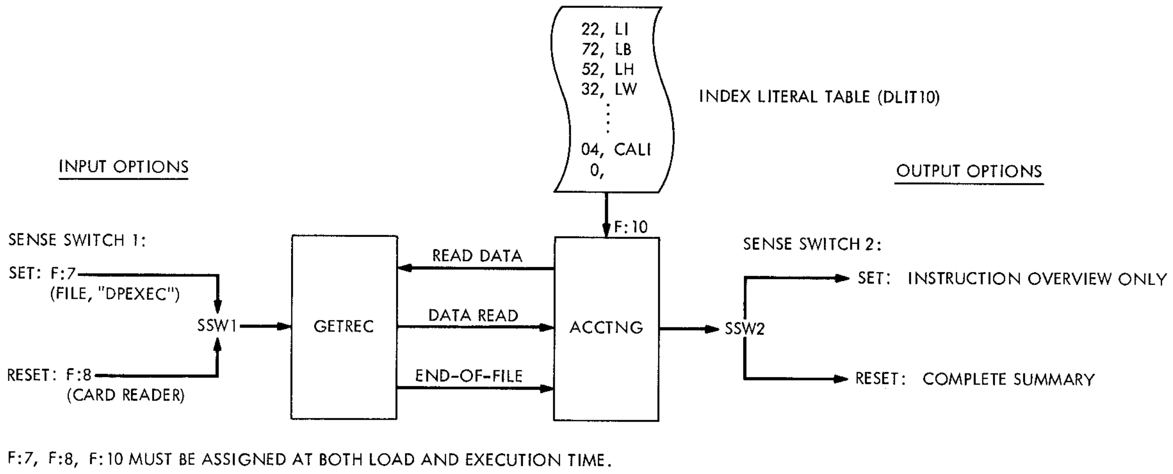


Fig. 10. Diagrammatic summary of ACCTNG